In the Claims:

1-72 (Cancelled)

73(Currently amended) A communication interface having n data lanes, said interface having a controller for sequentially and contiguously transmitting a header including a packet type field describing a payload data type, said controller generating a header distributed across a plurality of said data lanes, said controller also generating a variable amount of payload data comprising an encapsulated packet having an encapsulated header and encapsulated data, said payload data distributed sequentially across said n data lanes by said controller;

said encapsulated header containing information unrelated to said packet header other than said packet type field;

a field check sequence computed over the entire said payload data, concatenated to the end of said payload, and distributed sequentially across said n data lanes by said controller;

said header includes transmitting a START symbol on first said data lane, and the transmission of said payload

data is followed by said field check sequence distributed as bytes across said n data lanes and an END symbol on at least one said data lane;

said payload data includes transmitting successive data bytes canonically across said n successive data lanes up to data lane m, where m<=n;

an encoder coupled to each said data lane such that during intervals when said header or said payload is not being transmitted, each said encoder generates an alternating pattern of a first preamble symbol and a second preamble symbol distinct from said first preamble symbol ~~is transmitted~~ across said n data lanes;

and said n > 1.

74(Previously presented) The communication interface of claim 73 where said n is 4.

75(Previously presented) A process for transmitting data on a communications channel having a first, a second, a third, and a fourth data lane, each said data lane being 8 bits wide, said data comprising a header which includes a start symbol, payload type field, and variable length payload described by said payload type, said payload further having an encapsulated header and encapsulated payload, said variable length payload followed by a field check sequence

computed on said header and also said payload, said field check sequence spanning all said data lanes, the channel transmitting said data on successive clock intervals by sequentially placing said data on said first, said second, said third and said fourth data lane during a particular said clock interval, said process comprising the steps:

a first step of sending a synchronization symbol on all four said data lanes until said variable length payload is ready to be transmitted and not sending said synchronization symbol again until all after all said variable length payload is transmitted, said synchronization symbol being an alternating succession of a first preamble symbol followed by a second preamble symbol distinct from said first preamble symbol;

a second step of substantially simultaneously sending said header to said first data lane and part of said payload to the remaining three said data lanes during a first said clock interval;

a third step of incrementally transmitting the remainder of said payload data in a sequence of transmission events, each said transmission event occurring during a said successive clock interval and comprising sending said incremental payload data distributed across said four data lanes followed by said field check sequence until unsent

said field check sequence spanning one, two, or three lanes remains to be transmitted;

a fourth step of transmitting said unsent field check sequence by distributing it across said one, two, or three data lanes accompanied by an END symbol on one said data lane.

76(Previously presented) The process of claim 75 where no said unsent field check sequence remains and said END symbol is transmitted on said first data lane.

77(Previously presented) The process of claim 75 where no said unsent field check sequence remains and said END symbol is transmitted on said first data lane accompanied by said preamble transmitted on said second, said third, and said fourth data lanes.

78(Previously presented) The process of claim 75 where said unsent field check sequence is transmitted on said first said data lane and said END symbol is transmitted on said second data lane.

79(Previously presented) The process of claim 75 where said unsent field check sequence is transmitted on said first said data lane and said END symbol is transmitted on

said second data lane accompanied by said preamble transmitted on said third and said fourth data lanes.

80(Previously presented) The process of claim 75 where said unsent field check sequence is transmitted on said first and said second data lanes and said END symbol is transmitted on said third data lane.

81(Previously presented) The process of claim 75 where said unsent field check sequence is transmitted on said first and said second data lanes and said END symbol is transmitted on said third data lane accompanied by said preamble transmitted on said fourth data lane.

82(Previously presented) The process of claim 75 where said unsent field check sequence is transmitted on said first, second, and third said data lanes and said END symbol is transmitted on fourth said data lane.

83(Previously presented) The process of claim 75 where said unsent field check sequence is transmitted on said first, said second, and said third data lanes and said END symbol is transmitted on said fourth data lane accompanied by said preamble transmitted on said fourth data lane.

84(Previously presented) The process of claim 75 where each said clock rate is substantially 312.5Mhz.

85(Previously presented) The process of claim 75 where each said clock rate is 156.25Mhz and both a both positive edge and a negative edge are used to transfer said data.

86(Previously presented) The process of claim 75 where each said clock rate is 312.5Mhz and either a positive edge or a negative edge is used to transfer said data.

87(Previously presented) The process of claim 75 where each said data lane is encoded and serialized into a serial stream of data.

88(Previously presented) The process of claim 87 where said encoder is an 8B/10B encoder.

89(Previously presented) The process of claim 87 where said serial stream of data is transmitted as a differential electrical signal.

90(Previously presented) The process of claim 87 where said serial stream of data is transmitted as an optical signal.

91(Currently amended) A transmitter for sending data formed into ~~a stream~~ streams of 8-bit bytes by a controller, ~~the stream~~ said streams forming ~~comprising~~ a header followed by a variable length payload, said data substantially simultaneously transmitted on a first data lane, a second data lane, a third data lane, and a fourth data lane in a succession of time sequences in the following manner:

said controller sending a preamble on said first, said second, said third, and said fourth data lanes until said variable length data is ready to transmit, where said controller sending a preamble including sending the alternating sequence of a first preamble symbol and a second preamble symbol distinct from said first preamble symbol across said four data lanes, and when said data stream is ready to transmit:

said controller sending a START symbol on said first data lane and said first three successive bytes of data from said stream on said second, said third, and said fourth data lanes during one said time sequence;

said controller sending the remainder of said data stream by sending each subsequent four bytes of unsent data on said first, said second, said third, and said fourth data lanes during successive said time sequences until there is

insufficient data to send on all four said data lanes, said insufficient data being final data;

when there is no said final data to send, said controller sending said END symbol on said first lane, and said preamble on said second, said third, and said fourth lanes;

when said final data comprises one said data lane, said controller sending said final data on said first lane, an END symbol on said second lane, and said preamble on said third and said fourth lanes;

when said final data comprises two said data lanes, said controller sending said final data on said first and said second lane, an END symbol on said third lane, and said preamble on said fourth lane,

when said final data comprises three said data lanes, said controller sending said final data on said first, said second, and said third lane, and an end symbol on said fourth lane.

92(Previously presented) The transmitter of claim 91 where each said data lane is 8 bits wide.

93(Previously presented) The transmitter of claim 91 where each said data lane is 8 bits wide and is clocked at a rate of 312.5Mhz.

94(Previously presented) The transmitter of claim 93 where said 312.5Mhz clock comprises both the positive edge and the negative edge of a 156.25Mhz clock.

95(Previously presented) The transmitter of claim 93 where said 312.5Mhz clock comprises a positive edge or a negative edge of said 312.5Mhz clock.

96(Previously presented) The transmitter of claim 93 where each said data lane includes an encoder and a serializer, each said data lane generating a serialized stream of data.

97(Previously presented) The transmitter of claim 96 where each said data lane includes an encoder receiving data at said time sequence of substantially 312.5Mhz, and each said serializer is clocked at a rate of 10 times said encoder time sequence rate.

98(Previously presented) The transmitter of claim 96 where each said encoder uses 8B/10B encoding.

99(Previously presented) The transmitter of claim 93 where each said data lane comprises 8 bits of data and one

bit of clock, said clock operating at a rate of substantially 312.5Mhz.

100(Previously presented) The transmitter of claim 96 where data from each said data lane is transmitted least significant bit first and most significant bit last.

101(Previously presented) The transmitter of claim 96 where data from each said data lane is transmitted most significant bit first and least significant bit last.

102(Previously presented) A transmitter for generating four streams of serial data, said transmitter including:

a transmit buffer for receiving sequential data and a separator for separating said sequential data into four data lanes, said data having, in sequence, a header including a payload type field, a payload which includes an encapsulated header and encapsulated packet of a type described by said payload type field, and a field check sequence computed from said header and said payload, each said data lane comprising 8 bits of data and a clock operating at substantially 312.5Mhz;

said separator generating said four data lanes by prepending a START delimiter to the beginning of said

sequential data and appending an END delimiter to the end of said sequential data, thereafter forming a succession of four bytes of unsent sequential data and applying each of said four bytes of unsent sequential data to a particular said data lane, said four bytes of unsent sequential data applied at substantially the time;

each data lane having:

an encoder for converting said 8 bits of data accompanied by said clock into 10 bits of encoded data;

a serializer for transmitting said 10 bits of encoded data into a stream of serial data clocked at 10 times said encoder clock rate;

said encoder generating an alternating pattern of an even preamble symbol and an odd preamble symbol to indicate across said four data lanes when said START delimiter, said sequential data, and said END delimiter are not being transmitted.

103(Cancelled)

104(Previously presented) A receiver for receiving four streams of serial data and converting said four streams of serial data into a variable length packet, said receiver comprising:

four deserializers, each said deserializer coupled to a respective serial stream, each said deserializer converting said stream of serial data into 10 bits of encoded data accompanied by a clock for each said serial stream, said deserializer synchronizing to the alternating sequence of an first preamble symbol followed by an second preamble symbol distinct from said first preamble symbol;

four decoders, each said decoder coupled to a respective said deserializer output, each said decoder converting each said 10 bits of encoded data into 8 bits of decoded data, thereby producing 8 bits of decoded data accompanied by a clock;

an elasticity buffer coupled to each said 8 bit decoder data and decoder clock, said elasticity buffer receiving 8 bits of data from each decoder at a rate of substantially 312.5Mhz, and combining said decoder clock and data to form 32 bits of output data over successive intervals,

a packet generator coupled to said elasticity buffer output data and responsive to a START delimiter on a particular one of said four streams and an END delimiter on any said stream, where said END delimiter is accompanied by preamble symbols on at least one other stream, said packet generator forming said packet including a header, a payload, and a field check sequence by canonically concatenating data received from a first stream, second stream, third stream,

and fourth stream into said stream of 32 bits of data, said packet header containing a type field which identifies a particular type of said packet payload, said packet payload including an encapsulated header and an encapsulated payload;

where said packet header describes said packet payload type but does not include information derived from either said encapsulated header or said encapsulated payload of said packet payload.

105(Previously presented) The receiver of claim 104 where said decoder is an 8B/10B decoder.

106(Previously presented) The receiver of claim 104 where said variable length payload is formed using data received on the other three said decoders following a START symbol on one said decoder, thereafter using data from all four said decoders until receipt of an END symbol on any said decoder.

107(Previously presented) The receiver of claim 104 where said variable length payload is formed using data between a START symbol on one said decoder and an END symbol received on any said decoder.

108(Previously presented) The receiver of claim 104 where said elasticity buffer forms said variable length payload by concatenating data received from a first decoder, a second decoder, a third decoder, and a fourth decoder, where a START symbol is received on a first decoder and said variable length packet is formed from concatenating said data in sequence from said second decoder, said third decoder, said fourth decoder, and said first decoder, repeating until terminated by the receipt of an END symbol on any decoder.

109(Previously presented) The receiver of claim 104 where each said serial stream of data is derived from a differential electrical signal.

110(Previously presented) The receiver of claim 104 where each said serial stream of data is derived from an optical signal.

111(Previously presented) The receiver of claim 104 where said 312.5Mhz clock is the result of using both the rising edge and falling edge of a 156.25Mhz clock.

112(Currently amended) A process operative on a receive processor which generates ~~for generating~~ a variable length

packet from four streams of serial data, the process comprising:

deserializing each said serial stream into 10 bit encoded data, thereafter converting said 10 bit encoded data into four data lanes of 8 bit data, and forming a variable length packet as follows, said deserializer synchronizing said four data lanes using the alternating sequence of a first preamble symbol followed by a second preamble symbol distinct from said first preamble symbol, said alternating sequence present said four data lanes:

a first step of receiving a START symbol on said first data lane and said ordered variable length data on said second, said third, and said fourth data lanes during one said time sequence;

a second step of receiving the remainder of said variable length payload on said first, said second, said third, and said fourth data lanes during successive said time sequences until an END symbol is detected on one of said data lanes accompanied by payload data on at least one data lane and a preamble on at least one other data lane;

a third step of forming a variable length packet from said data from said START symbol to said END symbol, also maintaining the order of said data received on said first, said second, said third, and said fourth data lanes;

a fourth step of extracting a packet header including a packet type and a payload identified by said packet header type;

a fifth step of extracting an encapsulated header and an encapsulated packet from said payload according to said packet header type, where said packet header is unrelated to said extracted encapsulated header, and said packet header only identifies the type of said encapsulated header and said encapsulated packet.

113(Previously presented) The process of claim 112 where each said decoder is a 10B/8B decoder.

114(Previously presented) The process of claim 112 where each said 8 bit wide data lane is clocked at substantially 312.5Mhz.

115(Previously presented) The process of claim 112 where each said data lane is clocked at substantially 1/10th the rate of each said serial data.

116(Previously presented) The process of claim 114 where said 312.5Mhz clock comprises using either the rising edge or the falling edge of a 312.5Mhz clock.

117(Previously presented) The process of claim 114 where said 312.5Mhz clock comprises using both the rising and falling edge of a 156.25Mhz clock.

118(Previously presented) The process of claim 112 where each said serial stream of data is derived from a differential electrical signal.

119(Previously presented) The process of claim 112 where each said serial stream of data is derived from an optical signal.